



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/782,917	02/23/2004	Allen N. L. Lau	13527-2	7543

1059 7590 09/26/2007  
BERESKIN AND PARR  
40 KING STREET WEST  
BOX 401  
TORONTO, ON M5H 3Y2  
CANADA

EXAMINER

VU, TUAN A

ART UNIT PAPER NUMBER

2193

MAIL DATE DELIVERY MODE

09/26/2007

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

## Office Action Summary

Application No.

10/782,917

Applicant(s)

LAU ET AL.

Examiner

Tuan A. Vu

Art Unit

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 23 July 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1, 3 and 5-31 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1, 3, 5-31 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- ☐ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☒ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date 4/03/07.
- ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- ☐ Notice of Informal Patent Application
- ☐ Other: \_\_\_\_\_.

### DETAILED ACTION

1. This action is responsive to the Applicant's response filed 7/23/07.

As indicated in Applicant's response, claims 1, 3, 5-6, 11, 16, 18 have been amended; and claims 2, 4 canceled. Claims 1, 3, 5-31 are pending in the office action.

#### *Double Patenting*

2. A rejection based on double patenting of the "same invention" type finds its support in the language of 35 U.S.C. 101 which states that "whoever invents or discovers any new and useful process ... may obtain a patent therefor ..." (Emphasis added). Thus, the term "same invention," in this context, means an invention drawn to identical subject matter. See *Miller v. Eagle Mfg. Co.*, 151 U.S. 186 (1894); *In re Ockert*, 245 F.2d 467, 114 USPQ 330 (CCPA 1957); and *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970).

A statutory type (35 U.S.C. 101) double patenting rejection can be overcome by canceling or amending the conflicting claims so they are no longer coextensive in scope. The filing of a terminal disclaimer cannot overcome a double patenting rejection based upon 35 U.S.C. 101.

3. Claims 1, 3-13 are provisionally rejected under 35 U.S.C. 101 as claiming the same invention as that of claims 2-13 of copending Application No. 10,713,024 (hereinafter '024).

This is a provisional double patenting rejection since the conflicting claims have not in fact been patented.

As per instant claim 1, '024 claim 2 also recites 'unpacking the reference application... class files' and 'transforming the reference application... target mobile device' for the method of generating a target application from a Java 'reference application adapted to execute on a reference mobile device', the 'target application configured for a target mobile device'.

As per instant claims 3-13, '024 claims 3-13 also recite the same limitations which are worded substantially in an identical manner.

Art Unit: 2193

4. Claims 1, 3-18 are provisionally rejected under 35 U.S.C. 101 as claiming the same invention as that of claims 2-17 (respectively) of copending Application No. 10,975,346 (hereinafter '346). This is a provisional double patenting rejection since the conflicting claims have not in fact been patented.

**As per instant claim 1**, '346 claim 2 also recites method of generating a target application ('target application configured for a target mobile device') from a Java 'reference application *configured to* (as opposed to *adapted to*) execute on a reference mobile device', the method comprising: 'unpacking the reference application... class files'; and 'transforming the reference application... target mobile device' and transformation engine to modify reference application in bytecode. Even though '346 claim 2 presents a preamble shuffled in a slightly different word order, the intended semantic therein is identical to that of instant claim 1, the body of '346 claim 2 being identical to that of instant claim 1.

As per instant claims 3-17, '346 claims 3-17 also recite the same limitations, all of which worded in an identical manner.

**As per instant claim 18**, '346 claim 2 also recites a system for transforming Java reference applications ... target mobile device', comprising a transformation engine, a *plug-in* (which equivalent to claim 18 *plug-in*) the plug-in comprising instruction file and selected software code, 'wherein the transformation engine is adapted to access ... the selected software code' and wherein the reference application portion being transformed into target applications and wherein transformation engine is to modify the reference application bytecode.

***Claim Rejections - 35 USC § 101***

5. 35 U.S.C. 101 reads as follows:

Art Unit: 2193

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claims 18-31 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a “useful, concrete, and tangible result” be accomplished. An “abstract idea” when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the “useful arts” when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a “useful, concrete and tangible result”.

Specifically, claims 18-31 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The current focus of the Patent Office in regard to statutory inventions under 35 U.S.C. § 101 for method claims and claims that recite a judicial exception (software) is that the claimed invention recite a practical application. Practical application can be provided by a physical transformation or a useful, concrete and tangible result. The following link on the World Wide Web is for the United States Patent And Trademark Office (USPTO) policy on 35 U.S.C. §101.

[http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/guidelines101\\_20051026.pdf](http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/guidelines101_20051026.pdf)

Claim 18 recites a computer-implemented system adapted to execute mobile device applications, comprising a transformation engine adapted for computer execution, a plug-in interacting with an instruction file to modify portion of a software code. From the Specifications, the transformation engine is a Java application (para 0022 pg. 5), and a plug-in is known to be software, both of which described as being *adapted to* provide some functionality; hence said system clearly does not include a computer and in whole, the claim amounts to listing of software elements with software functionality without providing support of hardware components or hardware embodiment. Listing of mere Functional Descriptive material has been

Art Unit: 2193

addressed in the 101 Guidelines (refer to *guidelines101\_20051026.pdf*: Annex IV, pg. 51-52) as an inability to realize the functionality of the software and would be treated a non-statutory subject matter. That is, the claim does not reasonably convey that the recited software elements are able to perform any physical transformation, absent any hardware to carry out their functionality. As a whole, the claim amounts to listing of non-practical entities, i.e. insufficient to yield a final non-abstract result in terms of concrete, useful, and tangible result as required by the Practical Application Test; and is rejected for leading to a non-statutory subject matter. Claims 19-31, for not remedying to the hardware support deficiency, are also rejected as non-statutory subject matter.

***Claim Rejections - 35 USC § 102***

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

8. Claims 1-8, 10-15, 18-21, 24-28, 30-31 are rejected under 35 U.S.C. 102(e) as being anticipated by Ryzl, USPubN: 2003/0236657 (hereinafter Ryzl).

As per claim 1, Ryzl discloses a method of generating a target application from a reference application, the reference application being a Java application adapted to execute on a reference mobile device ( see para 0047-0049, pg. 4), the target application being configured for a target mobile device, the method comprising:

unpacking the reference application into a plurality of class files (see para 0051, pg. 5 – Note: unpacking *kjva.jar* of a EE interface package - see para 0047, pg. 4 – to change the looks into a configuration 70, Fig. 10 reads on unpacking a reference application of the EE);

transforming the reference application into the target application by a plug-in (see *pluggable architecture* – para 0046, pg. 4, para 0073, pg. 6; para 0060-0061, pg. 4; para 0066-0068, pg. 4 - Note: *pluggable architecture* and J2ME Apis reads on pluggable APIs in a wireless Java tool – see Fig. 9 -- for unpacking jar and for packing jar file to support MIDP and DoJa) , wherein the plug-in is adapted to transform a plurality of different reference applications into a corresponding plurality of target applications (See Fig. 20 – Note: EE default interfaces plus corresponding *Properties/Class* to be changed via the IDE tool into counterparts in a Emulator Configuration – see Fig 10-11, 15-17; Fig. 20 – reads on transforming reference applications into corresponding target applications) for a predetermined combination of the reference mobile device and the target mobile device;

wherein the reference application is in bytecodes during the transformation step ( see Forte<sup>™</sup> for Java<sup>™</sup> - Fig. 10; *J2ME ...verification step... bytecode*, para 0006, pg. 1 – Note: J2ME, CLDC and KVM – para 0013-0018, pg. 2; *scan ... bytecodes*, para 0015, pg. 2 – reads on wireless environment where package of Java are unpacked for integration by a Java tool that treats Java classes for interpreting, and executing via dynamic JVM compilation with code verifying of mobile emulator, hence handling Java bytecodes – Refer to ‘interpreter ... mobile device ...runs as an emulator’ of Instant Application, BACKGROUND, para 0008, pg. 2 or Admitted Prior Art).

**As per claim 3**, Rizl discloses wherein the plug-in comprises an instruction file (e.g. *MIDP ... configuration file* – para 0047, pg. 4; *properties file* – para 0051, pg. 4; *MIDP specification, adContent file* – para 0068, pg. 5; para 0071, pg 6) and at least one library (e.g. */lib/ext directory* – para 0051, pg. 4; *set of libraries* – para 0013-0014, pg. 2), wherein the transformation step comprises the instruction file instructing a transformation engine to modify a portion of the reference application with a selected software code stored in the library (e.g. Fig. 15-17; para 0071, pg. 6; Fig. 10; para 0073, pg. 6), wherein the portion of the reference application is not supported by the target mobile device.

**As per claim 5**, Rizl discloses adding a new class file (e.g. *AddataObject* - step 190 – Fig. 20) to the reference application ( Note: *addObject* to a default Emulator to obtain a changed Configuration 70 reads on *addind a new class*) .

**As per claim 6**, Rizl discloses one action selected from the group of: adding a new method, renaming an existing method, replacing a first object method call with a second object method call, replacing the first object method call with a static method call, renaming a constant pool entry, and inserting a new inner class to an existing class ( see Fig 15-17; refer to claim 5; *cut, copy, paste, delete, rename, save* -- para 0070, pg. 6).

**As per claims 7-8**, Rizl discloses saving the target application to a computer readable medium (step *ST144*, Fig. 17); and further comprising repeating step (a) and step (b) to transform the plurality of different reference applications into the plurality of corresponding target applications (See Fig. 20 – Note: pluggable architecture-based IDE tool by which EE default interfaces plus corresponding *Properties/Class* to be changed via the IDE tool into counterparts



in a Emulator Configuration – see Fig 10-11, 15-17; Fig. 20 – reads on transforming reference applications or plurality thereof into corresponding target applications or plurality thereof).

**As per claims 10-11**, Rizl discloses repackaging the target application into executable code (e.g. J2ME Wireless Compiler 184 – Fig 20; ST120 – Fig 14) wherein the repackaging step further comprises obfuscating (JAR – Fig. 17; ) the target the class files of the target application.

**As per claim 12-13**, Rizl discloses pre-verifying the class files (e.g. ST116 – Fig 14) of the target application; wherein the target application is a non-Java application (e.g. different applications types ... in a seamless manner – para 0049, pg. 4).

**As per claims 14-15**, Rizl discloses unpacking a JAR file (see para 0051, pg. 5 – Note: unpacking *kjva.jar* of a EE interface package - see para 0047, pg. 4 – to change the looks into a configuration 70, Fig. 10 reads on unpacking Jar of a reference application of the EE) of the reference application; and breaking an immutable image from the reference application into a plurality of smaller images (e.g. para 0018, pg. 2 – Note: unpacking a archive image reads on breaking it into a plurality of smaller images).

**As per claim 18**, Rizl discloses a computer-implemented system for transforming Java reference applications adapted to execute on a reference mobile device into corresponding target applications configured for a target mobile device, the system comprising:

a transformation engine adapted to execute on a computer; and a plug-in (see *pluggable architecture* – para 0046, pg. 4, para 0073, pg. 6; para 0060-0061, pg. 4; para 0066-0068, pg. 4) comprising:

an instruction file (e.g. *MIDP ... configuration file* – para 0047, pg. 4; *properties file* – para 0051, pg. 4; *MIDP specification, adContent file* – para 0068, pg. 5; para 0071, pg 6 ); and

Art Unit: 2193

a selected software code adapted to modify a portion of the reference application not supported by the target mobile device (e.g. Fig. 10; Fig 15-17; refer to claim 5; *cut, copy, paste, delete, rename, save* -- para 0070, pg. 6); wherein the transformation engine is adapted to instruct the computer to access the instruction file, the instruction file being adapted to direct the transformation engine to identify the portion of the reference application and to instruct the computer to modify the portion with the selected software code (e.g. *MIDP ... configuration file* -- para 0047, pg. 4; *properties file* -- para 0051, pg. 4; *MIDP specification, adContent file* -- para 0068, pg. 5; para 0071, pg 6) and at least one library (e.g. */lib/ext* directory -- para 0051, pg. 4; *set of libraries* -- para 0013-0014, pg. 2) to create the corresponding target application configured for the target mobile device (See Fig. 20 -- Note: EE default interfaces plus corresponding *Properties/Class* to be changed via the IDE tool into counterparts in a Emulator Configuration -- see Fig 10-11, 15-17; Fig. 20 -- reads on transforming reference applications into corresponding target applications),

wherein the transformation engine is adapted to modify the reference application in bytecode ( see Forte<sup>™</sup> for Java<sup>™</sup> - Fig. 10; *J2ME ... verification step... bytecode*, para 0006, pg. 1 -- Note: J2ME, CLDC and KVM -- para 0013-0018, pg. 2; *scan ... bytecodes*, para 0015, pg. 2).

**As per claim 19**, Rizl discloses wherein the instruction file comprises an XML file (e.g. para 0069, 0071, pg. 5, 6).

**As per claim 20**, Rizl discloses that the plug-in tool comprises a library, the library being adapted to store a plurality of the software codes (e.g. */lib/ext* directory -- para 0051, pg. 4; *set of libraries* -- para 0013-0014, pg. 2) adapted to modify a plurality of the portions.

As per claims 21, and 30-31, Rizl discloses that the transformation engine is a Java application (see Forte<sup>™</sup> for Java<sup>™</sup> - Fig. 10) wherein the target applications are Java applications (JAR classes – Fig. 17; para 0048, pg. 4; para 0066, pg.5 ) wherein the target applications are non-Java applications (refer to claim 13)

As per claims 24-26, Rizl discloses wherein the selected software code comprises a new method adapted for insertion into the reference application ( re claims 5-6); wherein the selected software code comprises a static method call adapted for insertion into the reference application (e.g. para 0068, para 0070, pg. 5); wherein the selected software code comprises a new method name adapted to replace a method name in the reference application ( re claim 6).

As per claims 27-28, Rizl discloses a new object method call adapted to replace (e.g. *rename* – para 0070, pg. 6; *change ... existing types* – para 0049 – Note: Java<sup>™</sup> pluggable architecture Forte – see Fig 10 -- equipped with GUI editing tool to replace existing applications reads on Java method call to replace an object) a method call in the reference application; a new class file adapted for insertion into the reference application (para 0066-0068, para 0070, pg. 5).

### ***Claim Rejections - 35 USC § 103***

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

10. Claims 9, 29 are rejected under 35 U.S.C. 103(a) as being unpatentable under Ryzl,

USPuBN: 2003/0236657.

As per claim 9, Ryzl does not explicitly disclose selecting a predetermined plug-in from a plurality of the plug-ins, the predetermined plug-in corresponding to the predetermined combination of the reference mobile device and the target mobile device, each of the plurality of the plug-ins corresponding to a different combination of the reference mobile device and the target mobile device. However, Ryzl teaches constant checking of jar file content to verify whether classes have been modified or still up-to-date to create the final Jar package ( see para 0061-0063, pg. 5) based on some predetermined manifest/descriptor information ( see Fig. 11, 14) or properties for a target mobile application based on the default configuration. The selecting and finally including of appropriate class for a given determined descriptor file entails a need to select not only class and method or properties but also the correct APIs or plug-ins to verify or retrieve such required objects as needed for the J2ME application ( see Fig. 20; *getAPIClassPath* - para 0047, pg. 4; *API ... profile ... meet the needs* – para 0016, pg 2 ). It would have been obvious for one skill in the art at the time the invention was made to implement the pluggable architecture by Ryzl so that it includes feature so as to enable selective (or as-needed) retrieval of plug-in APIs because these pluggable APIs being predetermined to support the retrieval, verification, and editing of objects being included in the final Jar package would enable the pluggable architecture of Ryzl's platform to dynamically effect call or invocations to integrate an application on a efficient basis approach which is contemplated for a mobile endeavor ( see J2ME, KVM, CLDC, MIDP, pg. 2), concerning which only suitable APIs would be needed to support a resource-constraint device (see para 0011-0013, pg 2).

As per claim 29, Rizl discloses a plurality of the plug-ins, each of the plurality of the plug-ins corresponding to a different combination of reference and target mobile device ( e.g.

para 0046, pg. 46; APIs para 0059-0063, pg. 5), but Ryzl does not explicitly disclose wherein the transformation engine is adapted to choose a selected one of the plug-ins corresponding to the different combination; however this selecting limitation has been addressed in claim 9 above.

11. Claims 16-17, 22-23 are rejected under 35 U.S.C. 103(a) as being unpatentable under Ryzl, USPubN: 2003/0236657; in view of Byman-Kivivuori et al., USPubN: 2004/0002305(hereinafter Byman)

**As per claims 16-17**, Rizl does not disclose wherein the transformation step comprises inserting an interception module into the reference application, wherein the interception module is adapted to intercept key events; wherein the transformation step comprises inserting a conversion table into the interception module, wherein the conversion table defines key-mapping from the reference mobile device to the target mobile device. Rizl discloses emulator interface with default properties and a IDE tool to implement a target configuration via modifying the default properties of the emulator interface ( see Fig.0047-0059, pg. 4-5; Fig. 11-13) in an integration and editing tool to implement executable application or interface properties according to a CLDC target. It was well-known that keypad function layout (and thereby patterns of keyboard input/event ) varies from one mobile device from one manufacturer to the next; and this is specified in a CLDC enabling a porting between platforms or different commerce applications; and such is suggested in Rizl's mention of the cellular phone expansion and their constraint in graphical capacity (Rizl: para 0011, pg. 2) with regard to which J2ME/CLDC provide as specification profile to support this mobile interface format integration ( see *implements EE interface* – para 0051-0053,pg 4; Fig. 15) to resolve interface differential as set

forth above. In a wireless network to provide WAP applications to mobile devices with deployment of JAR, class bytecodes based on a RFID or tags of a descriptor XML format file (Byman: para 0057, pg. 6) similar to the integration tool by Rizl, Byman discloses that the descriptor tags can designate the GUI elements including digits of a keyboard, such interface input susceptible to be replaced or augmented (Byman: para 0097-0098, pg. 11) and/or serve as indicative events that can be emulated via a descriptor tag for addressing a user request. In view of the descriptor-based mobile GUI-interface modification from one emulator configuration to a target configuration set forth above (Rizl: Fig.0047-0059, pg. 4-5; Fig. 11-13) in light of the J2ME<sup>™</sup> browser application (Rizl: J2ME ... web page ... awt.frame – para 0005, pg. 1) by Rizl, it would have been obvious for one skill in the art at the time the invention was made to provide Rizl's IDE tool with pluggable APIs to enable description of the key mapping or detection of keypad touching event as taught by Byman to handle differentials in mobile device manufacturer model user-input interface or request as concerned by Byman's descriptor emulation scheme. One would be motivate to provide event intercepting module/API into this IDE tool by Rizl to learn about the key input by the user given identification for each key as taught by Byman, and based on such key-mapping in view of the Byman's predefined descriptor or tags, handle the request of the user based on such keyboard touching event and learn from the descriptor mapping appropriate keyboard format pertinent the target mobile station in order to deliver appropriate services or deployed assets to the user as set forth by Byman ( Byman: para 0048, 0053-0054, pg. 5-6)

**As per claims 22-23, refer to rejections of claims 16-17 as set forth above.**

***Response to Arguments***

Art Unit: 2193

12. Applicant's arguments filed 7/23/07 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

**35 USC § 101 Rejection:**

(A) Applicants have submitted that 'engine adapted to execute on a computer' as amended to claim 18 would overcome the rejection; however, the above 'adapted to' does not explicitly convey that a computer is included with the system because this 'to execute' functionality is only a descriptive functionality, not having the weight of actually reciting or including a tangible hardware support per se. The claims will be rejected as set forth in the Office Action.

**35 USC § 102 Rejection:**

(B) Applicants have submitted that Ryzl does not teach a method of generating a target application from a reference application; but merely teaches compiling and executing emulators simulating various target devices; i.e. not transforming a reference application into a target application (Appl. Rmrks pg. 8, bottom half). The language of claim 1 amounts to 2 implementation steps: unpacking into a plurality of classes, then transforming by a plug-in reference applications into combination of target application for a mobile target device. The rejection has provided the cited parts of Rizl to map what constitutes 'unpacking' into classes and 'transforming' as recited into a plurality of mobile device applications. That is, the rejection has clarified what unpacking is mapped to (see para 0051, pg. 5 – Note: unpacking kjva.jar of a EE interface package - see para 0047, pg. 4 – to change the looks into a configuration 70, Fig. 10 reads on unpacking a reference application of the EE); and what said 'transforming' is analogized to (See Fig. 20 – Note: EE default interfaces plus corresponding *Properties/Class* to be changed via the IDE tool into counterparts in a Emulator Configuration – see Fig 10-11, 15-

17; Fig. 20 – reads on transforming reference applications into corresponding target applications). A reference application is viewed as the amount of reference furnished by Java source code packaged into an archive or library of reusable code, and the particular instance for specifying such reference application is being provided by an IDE language operable in a Forte tool by Rizl. When classes are unpacked and compiled by the tool in order to yield instances of code to be emulated for a given target platform, a transformation by the API and plug-in within the tool is disclosed. There is unclear teaching in the specifications that would have to be interpreted broadly; e.g. what *reference application* constitutes of? How is transforming done in terms of specifics? How unpacking (...into plurality of classes) justifies a bytecode nature and how this directly correlate with any bytecode stage of the application being modified?) For not providing sufficient details to preclude the above teachings by Rizl from reading onto the claimed limitations, the claim is deemed as fulfilled by the Rizl, notably in terms of the 2 step actions as set forth above. The argument is not sufficient to overcome the rejection, because as interpreted, the whole claim has been met (i.e. Rizl anticipates the claim).

(C) Applicants have submitted that Ryzl does not provide bytecode to bytecode transformation (Appl. Rmrks pg. 9, top) nor does Ryzl's emulator provide unpacking reference application into a plurality of class files. The limitation as to 'wherein the reference application is in bytecode during ... transformation step' is only interpreted as the application being compiled and modified during verification and assembling of classes by Rizl's Emulator/Forte tool such that during such stage code in bytecode form is further adjusted to reach a more compliant target object destined for the mobile devices' architecture or application requirements/standard. The claim does not exactly recite a transformation from 'bytecode to



Art Unit: 2193

bytecode', and such assertion is not given consideration necessarily because of the more plausible interpretation that the recited 'reference application' respective to the 'during the transformation step' appears to be just a package of class files being unpacked prior to compilation or to transformation; and when in such package-of-class-files form, it is not commonly accepted that compressed Java classes or archive Java file would be (actually and already) in bytecodes, unless the claim language expressly indicates or defines so. For the sake of argument, if the package were in bytecode format just as the Java classes are unpacked, sufficient teaching in the disclosure would have to support that, lest a 35 USC § 112 1<sup>st</sup> paragraph rejection be raised. However, the claim does not remotely enforce a 'bytecode to bytecode transformation' to preclude the Forte-based Java code verification by Rizl to read on the bytecode being transformed, as set forth in the rejection. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

(D) Applicants have submitted Rizl's emulating tool cannot correspond to unpacking the reference application (Appl. Rmrks, pg 9, 2<sup>nd</sup> para); but the weight or merit given to the limitations of claim 1 has been analyzed with proper interpretation in section B above, therefore the argument is deemed insufficient as set forth above.

(E) Applicants have submitted that Ryzl does not teach each and every element of claim 18; that is, not teaching 'instruction file' to 'direct transformation ... to identify ... to modify ... to create the corresponding target application' (Appl. Rmrks pg. 9, bottom). The above broadly claimed steps amount to using a tool to provide standard cut, paste, add, delete, update feature

Art Unit: 2193

typical of a editor with which Forte operates, and the steps of identifying, modifying to create the target application are but mere actions normally achieved in such environment. Since the claim as a whole amounts to the subject matter of claim 1, claim 18 has been addressed with the corresponding teachings by Rizl that have been applied similarly in claim 1. The response to the argument would have to include the effect of section B above. The argument is not sufficient to overcome the grounds of rejection against claim 18, as set forth in the Office Action; and the rejections by way of obviousness will also stand as a result of this.

In all, the claims are rejected as set forth in the Office Action.

### *Conclusion*

13. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

Art Unit: 2193

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 ( for non-official correspondence - please consult Examiner before using) or 571-273-8300 ( for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Tuan A Vu  
Patent Examiner,  
Art Unit 2193  
September 20, 2007